

# 计算机视觉实验——图像全景拼接

2021113117 王宇轩

## 1. 实验环境

- 操作系统: Windows
- 编程语言: Python

## 2. 文件列表

文件名	内容
sift.py	SIFT实现
ransac.py	RANSAC实现
main.py	主程序
实验报告.pdf	实验报告

## 3. 实验过程

### 3.1 SIFT算法提取特征点

实现的SIFT算法详见sift.py文件。在main.py中，通过

```
kp1, des1 = sift.computeKeypointsAndDescriptors(img1)
kp2, des2 = sift.computeKeypointsAndDescriptors(img2)
```

进行调用。

### 3.2 计算特征点匹配

```
# FLANN 参数设计
match = cv2.BFMatcher()
matches = match.knnMatch(des1, des2, k=2)
```

### 3.3 RANSAC算法消除误匹配点

实现的RANSAC算法详见ransac.py文件。在main.py中，通过

```
Max_num, H, inlier_points = ransac.ransac(good, kp1, kp2, confidence=4, iter_num=1000)
```

进行调用。

### 3.4 计算单应性矩阵

在ransac.py中实现：

```
def computeHomography(matches, kp1, kp2):
    point_map = np.array([
        [kp1[match.queryIdx].pt[0],
         kp1[match.queryIdx].pt[1],
         kp2[match.trainIdx].pt[0],
         kp2[match.trainIdx].pt[1]] for [match] in matches
    ])
    pairs = [point_map[i] for i in np.random.choice(len(point_map), 4)]
    A = []
    for x1, y1, x2, y2 in pairs:
        A.append([x1, y1, 1, 0, 0, 0, -x2 * x1, -x2 * y1, -x2])
        A.append([0, 0, 0, x1, y1, 1, -y2 * x1, -y2 * y1, -y2])
    A = np.array(A)

    # Singular Value Decomposition (SVD)
    U, S, V = np.linalg.svd(A)

    # V has shape (9, 9) for any number of input pairs. V[-1] is the eigenvector
    # of (A^T)A with the smallest eigenvalue. Reshape into 3x3 matrix.
    H = np.reshape(V[-1], (3, 3))

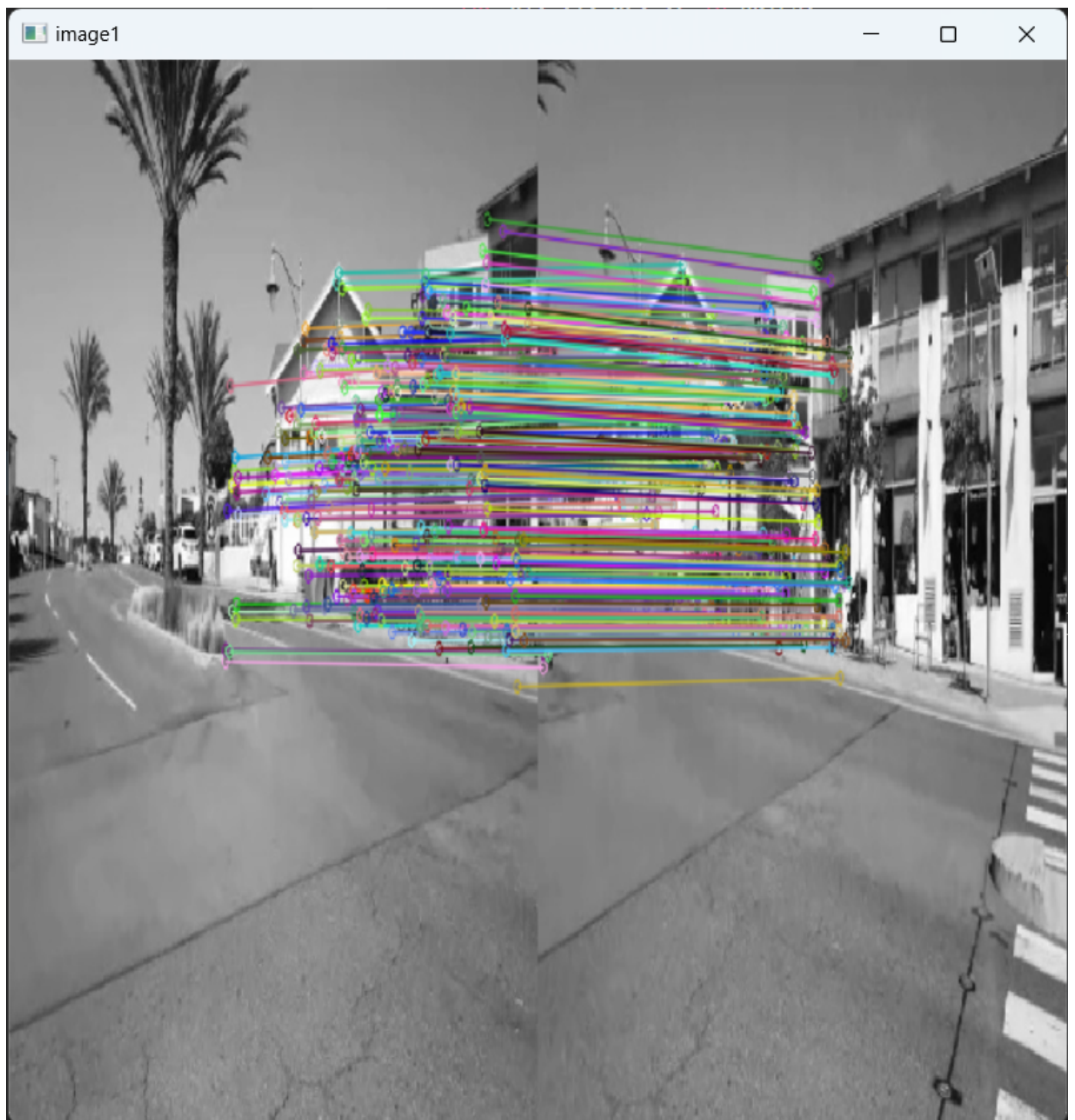
    # Normalization
    H = (1 / H.item(8)) * H
    return H
```

### 3.5 根据变换矩阵拼接图像

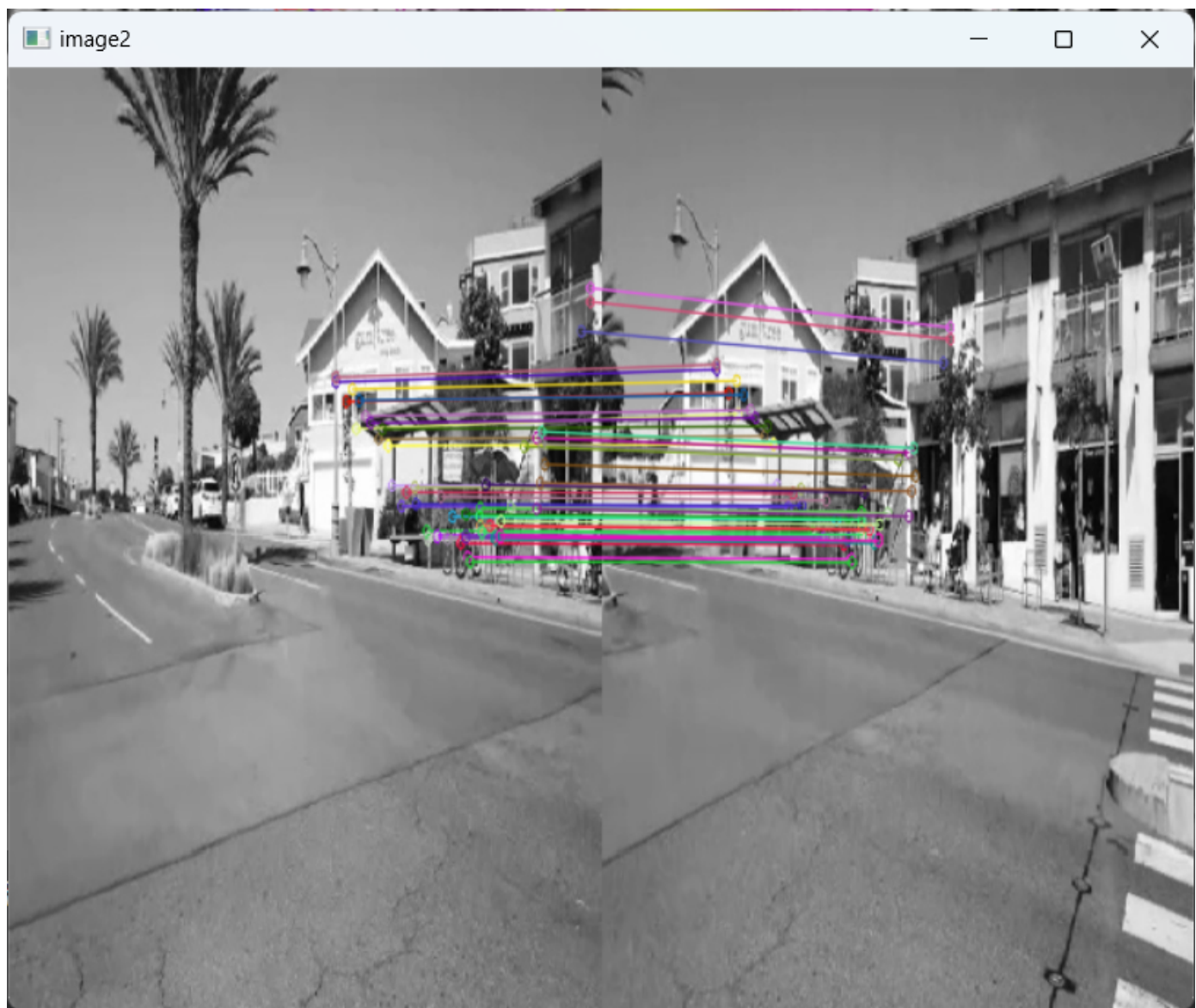
```
Panorama = cv2.warpPerspective(img2, H, (img2.shape[1] + img1.shape[1], img2.shape[0]))
```

## 4. 实验结果

初步计算的特征点匹配：



经RANSAC算法消除误匹配后的结果：



RANSAC得到的单应性矩阵H:

```
[[ 5.29760247e-01  5.85595236e-03  2.02615367e+02]
 [-2.47246335e-01  8.46691336e-01  4.06862256e+01]
 [-9.16902269e-04 -3.27926468e-06  1.00000000e+00]]
```

经H透视变换的右图:



拼接结果：

